

Mario Winter · Thomas Roßner · Christian Brandes · Helmut Götz

Basiswissen

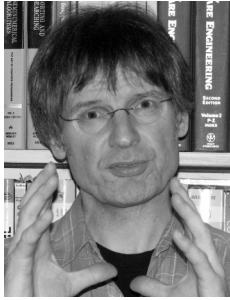


Modellbasierter Test

Aus- und Weiterbildung zum
ISTQB® Foundation Level – Certified Model-Based Tester

 dpunkt.verlag

2., vollständig überarbeitete und aktualisierte Auflage



Mario Winter ist Professor am Institut für Informatik der Technischen Hochschule Köln und dort Mitglied des Forschungsschwerpunkts »Software-Qualität«. Er ist Gründungsmitglied des German Testing Board e.V. und war von 2003 bis 2011 Sprecher der GI-Fachgruppe »Test, Analyse und Verifikation von Software«. Seine Lehr- und Forschungsschwerpunkte sind Softwareentwicklung und Projektmanagement, insbesondere die modellbasierte Entwicklung und Qualitätssicherung von Software. Er ist Autor und Mitautor zahlreicher Publikationen im Bereich Softwareentwicklung und Softwaretest, u.a. der Fachbücher »Der Integrationstest« und »Praxiswissen Softwaretest – Testmanagement« sowie der iX-Studie »Software-Testmanagement«.



Thomas Roßner ist Mitglied des Vorstands der imbus AG. Er verfügt über jahrelange praktische Erfahrung in der Planung und Entwicklung von Softwaretests in verschiedenen Branchen und Technologien. Er leitete nationale und internationale Forschungs- und Entwicklungsprojekte zum modellbasierten Testen sowie zur Kosten-Nutzen-Optimierung von Testprozessen. Der ISTQB® Certified Tester (Advanced Level Testmanager) ist außerdem SPICE- und TestSPICE-Spezialist. Zudem ist er Mitautor der iX-Studien »Software-Testmanagement« und »Modellbasiertes Testen« sowie des Fachbuches »Praxiswissen Softwaretest – Testmanagement«.



Christian Brandes ist Trainer und Principal Consultant bei der imbus AG. Der promovierte Mathematiker verfügt über langjährige Projekterfahrung als Testmanager, Testarchitekt, Testdesigner und Testprozessberater. Er ist ISTQB® Certified Tester (Full Advanced Level, Agile Tester Extension) und als Hochschuldozent für Softwaretest tätig. Zahlreiche Publikationen und Vorträge – zu modellbasiertem Testen, agilem Testen, Testautomatisierung und testbaren Architekturen – runden sein Portfolio ab.



Helmut Götz ist Senior Key Expert innerhalb der zentralen Forschungsabteilung der Siemens AG in Erlangen. Seinen Themenschwerpunkt hat er seit fast 20 Jahren im Bereich der Softwarequalitätssicherung sowie in der Etablierung ganzheitlicher Testansätze bei Produktfamilien und Systemen innerhalb der Siemens AG. Als Senior-Key-Experte für »MBT für komplexe Systeme« treibt er die Forschungs- und Kollaborationsaktivitäten auf diesem Gebiet voran. Neben zahlreichen Konferenzpublikationen ist er Mitautor der iX-Studie »Modellbasiertes Testen«.

Mario Winter · Thomas Roßner · Christian Brandes · Helmut Goetz

Basiswissen modellbasierter Test

**Aus- und Weiterbildung zum
ISTQB® Foundation Level –
Certified Model-Based Tester**

2., vollständig überarbeitete und aktualisierte Auflage



dpunkt.verlag

Mario Winter – mario.winter@th-koeln.de
Thomas Roßner – thomas.rossner@imbus.de
Christian Brandes – christian.brandes@imbus.de
Helmut Götz – helmut.h.goetz@siemens.com

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Birgit Bäuerlein
Herstellung: Susanne Bröckelmann
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:
Print 978-3-86490-297-0
PDF 978-3-96088-005-9
ePub 978-3-96088-006-6
mobi 978-3-96088-007-3

2., vollständig überarbeitete und aktualisierte Auflage
Copyright © 2016 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort zur zweiten Auflage

Modellbasiertes Testen (MBT) umfasst die Nutzung von Modellen für die Automatisierung von Testaktivitäten sowie die Modellierung von Artefakten im Testprozess. In der Literatur sind mittlerweile Hunderte von Veröffentlichungen zu finden, die zunächst vornehmlich in wissenschaftlichen Konferenzen und Zeitschriften publiziert wurden. Seit ca. 2005 erschienen zunehmend auch anwendernahe bzw. praxisbezogene Artikel zum modellbasierten Test und inzwischen sind bereits mehrere englischsprachige Fachbücher zum Thema verfügbar. Im März 2009 erschien die Werkzeugstudie »Modellbasiertes Testen« [Götz 2009] als erstes einschlägiges deutschsprachiges Werk zu MBT. Im Mai 2010 folgte dann die erste Auflage des hier vorgelegten Buches »Basiswissen modellbasierter Test«.

Fast sechs Jahre sind seither ins Land gegangen. Haben diese sechs Jahre das seinerzeit als sehr innovativ, ja sogar experimentell geltende Thema MBT vorangebracht?

Tatsächlich hat sich das modellbasierte Testen immer weiter von der Schlagwortebene und Grundlagenforschung hin zur praktischen industriellen Anwendung entwickelt. Doch viele der Herausforderungen, die wir seinerzeit identifiziert und erläutert haben, sind auch heute noch aktuell:

- MBT ist schwierig einzuführen. Der Testprozess muss dafür eine ausreichende Reife aufweisen, sonst führt MBT eher zu neuen Problemen, als dass es zur Lösung von bestehenden beiträgt. Vielfach lässt sich beobachten, dass viele Testteams und -organisationen nach wie vor mit grundlegenden Problemen wie z.B. schlecht formulierten, veralteten oder gar nicht vorhandenen Anforderungen, einem inadäquaten Testprozess und knapp bemessenen Testzeiträumen kämpfen.
- Für eine effiziente Nutzung von MBT müssen Entwicklungs- und Testteam sowie fachliche Anforderer nicht nur an einen Tisch, sondern in eine gemeinsame Arbeitsumgebung, bestehend aus Model-

Was ist modellbasiertes Testen?

Modellbasiertes Testen – von Schlagwortebene und Grundlagenforschung hin zu praktischer industrieller Anwendung

lierungssprache und -werkzeug, gebracht werden. Die Organisationsstruktur vieler Unternehmen stellt auch heute noch ein Hindernis hierfür dar. Allerdings lässt z.B. die DevOps-Bewegung die Hoffnung aufkeimen, dass die Voraussetzungen für ein gemeinsames Arbeiten an Testmodellen besser werden.

- Die Auswahl an Werkzeugen und Modellierungssprachen sowie deren Einbindung in typische Entwicklungs- und Testwerkzeuglandschaften ist besser geworden. Dennoch werden MBT-Werkzeuge noch lange nicht als so benutzerfreundlich wahrgenommen wie z.B. Testmanagementwerkzeuge, und die Hürde für ihren produktiven Einsatz ist nach wie vor hoch. Ein Beleg hierfür lässt sich beispielsweise unter [URL: Weilkien's] nachlesen.

Dennoch findet man immer häufiger Berichte über den erfolgreichen Einsatz von MBT in der Softwareindustrie. Auch in der Lehre wird das Thema zunehmend adressiert und zumindest auf Schlagwortebene erfreut sich MBT inzwischen eines deutlich höheren Bekanntheitsgrades als 2010.

Zu Letzterem hat unter anderem auch die Aktivität des International Software Testing Qualifications Board (ISTQB®) beigetragen, das modellbasierte Testen in den Lehrplan des ISTQB® Certified Tester als Add-on aufzunehmen. Neben einer höheren Marktdurchdringung sorgt dieser Ende 2015 erschienene Lehrplan auch für eine gewisse Vereinheitlichung von Begriffen, Sichtweisen und Techniken von MBT.

Als wir die ersten Versionen dieses Lehrplans sichteten, waren wir sehr erfreut, dass durch die MBT-Arbeitsgruppe des ISTQB® weite Teile der Inhalte unseres Buches aufgegriffen und somit quasi »zu offiziellen Weihen« geführt wurden. Daher ist es uns vergleichsweise leichtgefallen, mit der nun vorliegenden zweiten Auflage eine Überarbeitung herauszubringen, die die Lernziele des MBT-Lehrplans vollständig abdeckt.

Wir haben uns jedoch bewusst dafür entschieden, den Aufbau des Buches nicht am Lehrplan auszurichten, sondern die Struktur der ersten Auflage weitgehend beizubehalten. Auch haben wir Inhalte, die über den Lehrplan hinausgehen, nicht aus dem Buch entfernt. Stattdessen haben wir eine Leseanleitung ergänzt, die diejenigen Leser, die sich gezielt auf die Prüfung zum »ISTQB® Certified Model-Based Tester« vorbereiten wollen, zu den relevanten Stellen des Buches leitet.

Wir, die Autoren, verfolgen mit dem vorliegenden Buch (wie schon mit der 1. Auflage) das Hauptziel, Ihnen bei der Erschließung des modellbasierten Testens ein guter Ratgeber zu sein! Wir sind überzeugt davon, dass Sie mehr Freude daran haben werden, gute Modelle für gutes modellbasiertes Testen zu erstellen, als die vielen benötigten Test-

fälle bzw. Testskripte für gutes manuelles bzw. automatisiertes Testen zu schreiben. Daher möchten wir Sie mit dem Buch in die Lage versetzen,

- die grundsätzlichen Prinzipien und Methoden des modellbasierten Testens verstehen und charakterisieren zu können,
- den aktuellen Stand der Technik des modellbasierten Testens zu überblicken und die technische und wissenschaftliche Originalliteratur lesen sowie neue Vorschläge hinsichtlich der Relevanz für Ihre Projekte einschätzen zu können und
- die angebotenen Werkzeuge hinsichtlich der zugrunde liegenden Verfahren analysieren und eine Vorauswahl auf der Basis Ihres eigenen Projektkontexts treffen zu können.

Das Buch bietet Ihnen eine breite Einführung in das Thema, nicht aber werkzeugspezifische Lösungen für konkrete Testprobleme. Es stellt die Grundlagen modellbasierten Testens dar und gibt einen Überblick über verschiedene Modellierungssprachen und Testverfahren. Es zeigt auf, welche Voraussetzungen für den Einsatz modellbasierten Testens in einer Organisation zu schaffen sind, welche möglichen Fallstricke eine erfolgreiche Einführung verhindern können und wie man diese Fallstricke vermeidet. Anhand von einfachen Beispielanwendungen aus verschiedenen Industrieinsätzen wird insbesondere auch auf die Qualitätsgewinne und Einsparungspotenziale eingegangen, die durch MBT möglich sind.

Die Zielgruppe dieses Buches ist breit angelegt – schließlich soll ein »Basiswissen«-Buch die behandelten Themen so darstellen, dass nicht nur Testspezialisten einen Nutzen davon haben. Das Buch richtet sich primär an

- Tester bzw. (technische) Testanalysten sowie Entwickler mit Interesse an modernen modellbasierten (Test-)Ansätzen und
- Testmanager mit der Notwendigkeit zu effizienterem Testen.

Im Vordergrund stehen dabei solche Tester, die bisher eher nur eine diffuse Vorstellung davon haben, was MBT konkret sein könnte und wie sie es selbst möglicherweise einsetzen könnten. Insbesondere Testanalysten, die sich mit dem methodischen Übergang von zu testenden Anforderungen (z.B. in Modellform) hin zu Testfällen beschäftigen, werden sich angesprochen fühlen. Aber auch Testmanager, die im Rahmen ihrer Teststrategie die ihnen zur Verfügung stehenden Ressourcen möglichst nutzbringend einsetzen wollen, werden Hinweise zum Kosten-Nutzen-Verhältnis modellbasierter Ansätze finden. Testanalysten und technische Testanalysten werden je nach Modellierungshinter-

grund Möglichkeiten entdecken, ihre Testfälle in Modellform aufzubereiten.

Ebenso sollen auch Projektleiter, Anforderungsanalytiker, Modellierer und Entwickler erfahren, welche Möglichkeiten modellbasierte Teststrategien zu bieten haben – schließlich genügen oft schon geringe Erweiterungen oder Anpassungen an vorhandene Modelle, um für den Test relevante Artefakte aus ihnen »auf Knopfdruck« erzeugen zu können (wobei der »Knopf« in der Regel erst konzipiert und realisiert werden muss).

Schließlich will das Buch generell allen, die wissen möchten, was es mit dem Begriff »MBT« denn nun auf sich hat, eine Antwort geben. Es kann als Leitfaden zur Entwicklung einer eigenen MBT-Methodik und als Entscheidungsgrundlage für deren Einsatz herangezogen werden.

Die einzige Voraussetzung zum »erfolgreichen« Lesen des Buches sind Grundkenntnisse des Softwaretestens, wie sie z. B. im Rahmen des »ISTQB® Certified Tester – Foundation Level« vermittelt werden. Der dort beschriebene fundamentale Testprozess wird im vorliegenden Buch auf modellbasierte Disziplinen erweitert bzw. angepasst. Auch die dort vorgestellten Test- und Testmanagementverfahren werden – soweit sie für MBT relevant sind – aufgegriffen und in den MBT-Kontext eingeordnet.

Danksagung

Zu allererst gebührt unser Dank unseren Familien. Die Erstellung eines Buchmanuskripts ist nicht familienkompatibel, da man Ruhe benötigt und – viel schwieriger – mehrere Stunden Zeit »am Stück«. Keiner der Autoren konnte dafür eine längere Auszeit nehmen, sondern das Buch musste von Anfang an parallel zum Tages- und Projektgeschäft entstehen. Dies setzt ein großes Maß an Toleranz bei allen unmittelbar Betroffenen voraus, und wir hatten das große Glück, dieses Maß sehr oft zu erfahren.

Weiterhin danken wir dem dpunkt.verlag, allen voran Frau Preisendanz, die uns trotz mehrfacher Terminverschiebungen (einige davon höherer Gewalt wie Krankheitsausfällen geschuldet) die Treue hielt. Wir hoffen, dass das Urteil der Leser sie im Nachhinein in ihrer Geduld bestätigen wird.

Wir danken ebenso den Lesern der ersten Auflage, die uns mit ihren wertvollen Hinweisen an vielen Stellen aus dem sich fast zwangsläufig ergebenden »Wald«, den wir »vor lauter Bäumen« nicht mehr gesehen haben, geführt haben.

Und schließlich danken die Autoren einander für das funktionierende Kollektiv, das konstruktive Diskutieren auch abweichender

Standpunkte und das Vergnügen, am selben Strang zu ziehen. Die Diskussionen rund um den vom ISTQB® veröffentlichten Lehrplan und das Vorgehen bei der Gestaltung dieser Neuauflage waren – genau wie vor 6 Jahren – gehaltvoll, konstruktiv und unverändert von einer spürbaren Begeisterung für das Thema MBT geprägt.

Wir wünschen unseren Lesern, dass das Buch ihre Erwartungen erfüllt und sie es mit Gewinn und Vergnügen zurate ziehen.

*Mario Winter, Thomas Roßner, Christian Brandes, Helmut Götz,
im Juli 2016*

Geleitwort zur ersten Auflage

von Prof. Dr. Ina Schieferdecker
TU Berlin/Fraunhofer FOKUS

Modellbasiertes Testen geistert seit Jahren durch Konferenzen, Foren und Diskussionsrunden. In einer Vielzahl von Forschungs- und Entwicklungsprojekten sind Arbeiten mit dem Ziel durchgeführt worden, unter Nutzung von Modellen das Testen von softwarebasierten Systemen effektiver und effizienter zu gestalten. Dabei soll sowohl der Aufwand für das Testen minimiert als auch die Qualität der Tests erhöht werden. Die Kernidee des modellbasierten Testens wurde bereits mit der Softwarekrise in den 1960ern formuliert und ist anhand automatenbasierter Ansätze zumindest theoretisch umgesetzt worden. Jedoch bieten erst heutige Modellierungs- und Automatisierungsmethoden und -werkzeuge entlang der Model-Driven-Engineering-Initiativen Möglichkeiten, die Kernideen in den breiten praktischen Einsatz für softwarebasierte Systeme industrieller Größe zu überführen.

Nur: Was ist genau mit modellbasiertem Testen gemeint? Arbeitet man sich in dieses Gebiet ein, so ergeben sich umgehend verschiedene Sichten, die die Breite, das Potenzial, aber auch die Herausforderungen beim modellbasierten Testen erahnen lassen: Welche Modelle sind gemeint? Systemmodelle, Anforderungsmodelle, Entwurfsmodelle, Nutzermodelle, Architekturmodelle, Verhaltensmodelle etc.? Oder gar Testmodelle? Was ist überhaupt ein Modell? Welche Techniken werden für die Modellierung genutzt? Auf welcher Abstraktionsstufe wird modelliert? Welche Aspekte werden modelliert? Welche Phasen im Testprozess werden durch einen modellbasierten Ansatz tangiert? Geht es im Wesentlichen um einen modellbasierten Testentwurf inklusive der automatisierten Generierung der Testfälle? Oder aber geht es auch um die Automatisierung der Testausführung? Ist eine schrittweise Einführung modellbasierter Testansätze möglich? Und wenn ja – wie? Und wie entwickelt sich modellbasiertes Testen langfristig, wenn das softwarebasierte System weiterentwickelt wird? Wie bettet sich die Weiterentwicklung der modellbasierten Tests in diese Wartung und Weiterentwicklung ein? Usw. usf., bis hin zur »K.-o.-Frage«: Lohnt

sich das Ganze überhaupt? Oder aber bis zu einer anderen K.-o.-Frage: Machen wir das nicht eh schon seit Langem?

Lassen Sie sich mit diesem Buch in das Gebiet einführen und erwarten Sie eine kritische Diskussion und Reflektion obiger und weiterer Fragen. Während es eine Vielzahl englischsprachiger Publikationen und Abhandlungen zum modellbasierten Testen gibt, lag ein vergleichbares Werk für den deutschsprachigen Raum noch nicht vor. Thomas Roßner, Christian Brandes, Helmut Götz und Mario Winter haben sich der Aufgabe gestellt, das Gebiet des modellbasierten Testens zu strukturieren, darzustellen und beispielhaft zu erläutern. Dabei wird aufbauend auf einer allgemeinen Betrachtung von Modellierungs- und Testansätzen ein systematischer Zugang zum modellbasierten Testen gegeben: Angefangen mit Modellkategorien und verschiedenen Arten modellbasierten Testens wird die Einbettung in den Testprozess erläutert und die Qualität der Modelle diskutiert. Kapitel zur Automatisierung der Testgenerierung, der Testausführung und zur Klassifikation von Werkzeugen für das modellbasierte Testen unterlegen die konzeptionellen Betrachtungen mit praktischen Herangehensweisen. Anwendungsbeispiele reflektieren im Buch durchgängig die präsentierten Ansätze. Abschließend werden Migrationsstrategien, Effektivitätsbetrachtungen sowie Vor- und Nachteile modellbasierten Testens kritisch diskutiert, sodass eine objektivierte Analyse modellbasierter Testmethoden für eigene Entwicklungs- und Qualitätssicherungsprozesse ermöglicht wird.

Wenn ich eine Kritik ansetzen darf, dann insbesondere die, warum das Buch erst jetzt erscheint, haben die Autoren doch über Studien, Artikel und Vorträge das Gebiet des modellbasierten Testens schon längst durchdrungen. Andererseits, gut Ding will Weile haben. Das Buch überzeugt in seinem stilistisch wohlausgewogenen Herangehen an die Materie, die aus verschiedenen Perspektiven betrachtet und analysiert wird. Das Buch präsentiert Basiswissen zum modellbasierten Testen. Eine Fortsetzung wäre wünschenswert, in der die diversen Methoden anhand konkreter Techniken und Algorithmen noch detaillierter erörtert werden würden.

Ina Schieferdecker
Juni 2010

Inhaltsübersicht

Teil I	MBT – Einstieg und Grundlagen	1
1	Einleitung	3
2	Testen heute	23
3	Einführung in die Modellierung	47
4	MBT – ein Einstieg	105
5	Intermezzo 1: Türsteuerung	119
Teil II	MBT im Testprozess	133
6	MBT und der Testprozess	135
7	Planung und Steuerung	169
8	Intermezzo 2: CarKonfigurator	197
9	Modellierung und Modellprüfung	213
10	Testauswahlkriterien und Testfallgenerierung	265
11	Realisierung, Durchführung und Auswertung	311
Teil III	MBT im Praxiseinsatz	341
12	Werkzeuge für MBT	343
13	MBT erfolgreich einführen	353
14	Wirtschaftlichkeitsbetrachtung	383
15	Möglichkeiten und Grenzen von MBT	403

Anhang	417	
A	Abkürzungsverzeichnis	419
B	Glossar	421
C	Quellen	431
	Index	445

Inhaltsverzeichnis

Teil I	MBT – Einstieg und Grundlagen	1
1	Einleitung	3
1.1	Model Based Testing – nur ein Hype?	3
1.2	Viele Definitionen für modellbasiertes Testen	7
1.3	Ziele des Buches	10
1.4	Aufbau des Buches	13
1.5	Leseanleitung für die Qualifikation zum ISTQB® Certified Model-Based Tester	15
1.6	Zwei Fallbeispiele	20
1.6.1	Produktskizze CarKonfigurator	20
1.6.2	Produktskizze Türsteuerung	21
2	Testen heute	23
2.1	Grundbegriffe von Qualitätssicherung und Testen	23
2.1.1	Testverfahren	26
2.1.2	Kriterien zur Testüberdeckung	27
2.1.3	Teststufen	28
2.1.4	Randbedingungen des Testens	31
2.2	Der fundamentale Testprozess	32
2.2.1	Phasen des Testprozesses	33
2.2.2	Testplanung	34
2.2.3	Teststeuerung und -kontrolle	34
2.2.4	Testanalyse und -entwurf	35
2.2.5	Testrealisierung und -durchführung	35
2.2.6	Bewertung von Endkriterien und Bericht	36
2.2.7	Abschluss der Testaktivitäten	37
2.2.8	Rollen im Testprozess	37

2.3	Herausforderungen beim Testen heute	38
2.3.1	Kostenfaktor Test	39
2.3.2	Mangelnde Testqualität	40
2.3.3	Tester mit Fach-, aber ohne IT-Wissen	41
2.3.4	Komplexität der Tests	42
2.3.5	Testautomatisierung	43
2.3.6	Mangelnde Qualität der Testbasis	43
2.3.7	Zu späte Einbindung des Endkunden	44
2.4	Zusammenfassung	45
3	Einführung in die Modellierung	47
3.1	Wat is'n Modell?	47
3.2	Allgemeiner Vergleich von Texten und Modellen	50
3.2.1	Merkmale und Mängel natürlicher Sprache	50
3.2.2	Vorteile formaler Modelle und visueller Darstellungen	51
3.3	Exkurs: Grundbegriffe der Graphentheorie	54
3.4	UML – ein Standard der Softwaremodellierung	57
3.5	Modellierung statischer Strukturen	58
3.5.1	Objektmodellierung	58
3.5.2	Klassenmodellierung	60
3.5.3	Pakete und Komponenten	66
3.5.4	ER-Modellierung	68
3.6	Modellierung dynamischen Verhaltens	69
3.6.1	Aktivitätsmodellierung	70
3.6.2	Geschäftsprozessmodellierung mit BPMN	74
3.6.3	Zustandsmodellierung	76
3.6.4	Interaktionsmodellierung	81
3.7	Nebenläufigkeits- und Echtzeitmodellierung	85
3.7.1	Petri-Netze	85
3.7.2	Zeitdiagramm	88
3.8	Umgebungs-, Funktions- und Nutzungsmodellierung	89
3.8.1	Umgebungsmodellierung mit Akteuren	89
3.8.2	Funktionsmodellierung mit Anwendungsfällen	90
3.8.3	Nutzungsmodellierung	94
3.8.4	Präzisierung von Modellen mit der OCL	97
3.9	Metamodellierung und Profile	98
3.9.1	UML – Spracharchitektur und Metamodell	99
3.9.2	Stereotype und Profile	100
3.10	Zusammenfassung	103

4	MBT – ein Einstieg	105
4.1	Detaillierte Definition von MBT	105
4.2	MBT zur Testfallgenerierung	106
4.3	MBT im Projektkontext	107
4.4	MBT-Modellkategorien	109
4.4.1	Drei Modellkategorien für MBT	109
4.4.2	Umgebungsmodelle	110
4.4.3	Systemmodelle	111
4.4.4	Testmodelle	112
4.5	Nutzen von MBT	114
4.5.1	MBT reduziert Fehler in frühen Entwicklungsphasen ...	114
4.5.2	MBT liefert mehr und bessere Testfälle	115
4.5.3	MBT unterstützt die Testautomatisierung	116
4.6	Einordnung von MBT in den Softwaretest	117
4.7	Zusammenfassung	118
5	Intermezzo 1: Türsteuerung	119
5.1	Vorüberlegungen und Planung	119
5.2	Modellierung	
	Schritt 1 – funktionale Anforderungen	119
5.2.1	Strukturmodell	120
5.2.2	Verhaltensmodell	121
5.3	Generierung von funktionalen Testfällen	122
5.4	Modellierung Schritt 2 – Sicherheitsanforderungen	125
5.5	Neugenerierung der Testfälle	127
5.6	Bewertung des gewählten MBT-Vorgehens	130
5.7	Zusammenfassung	131
Teil II	MBT im Testprozess	133
6	MBT und der Testprozess	135
6.1	Ausprägungen von MBT im Testprozess	136
6.1.1	Modellorientiertes Testen	136
6.1.2	Modellgetriebenes Testen	138
6.1.3	Modellzentrisches Testen	139
6.2	Allgemeine Integration von MBT in den Testprozess	141

6.3	Abhängigkeit des MBT-Prozesses von den verwendeten Modellkategorien	144
6.3.1	Übersicht über die MBT-Varianten	144
6.3.2	Ausschließlich testmodellgetriebene MBT-Prozesse	145
6.3.3	Ausschließlich systemmodellgetriebene MBT-Prozesse ..	150
6.3.4	Das Systemmodell als Ausgangspunkt für ein generiertes Testmodell	153
6.3.5	System- und testmodellgetriebene MBT-Prozesse	155
6.4	Einsatz von MBT für verschiedene Testziele und Teststufen	155
6.4.1	Komponententest	157
6.4.2	Integrationstest	157
6.4.3	Systemtest	158
6.4.4	Systemintegrationstest	158
6.4.5	Abnahmetest	159
6.5	MBT und der Entwicklungsprozess	159
6.5.1	Allgemeine Aufgaben bei der Einbettung von MBT in den Entwicklungsprozess	159
6.5.2	Betrachtung verschiedener Vorgehensmodelle im Zusammenhang mit MBT	160
6.6	Zusammenfassung	167
7	Planung und Steuerung	169
7.1	Risikoidentifikation und -analyse	170
7.2	Bestimmung der Teststrategie	172
7.3	Aktivitätenplanung und Aufwandsschätzung	174
7.3.1	Aktivitäten und Artefakte	174
7.3.2	Aufwandsschätzung	176
7.4	Werkzeuge und Infrastruktur	180
7.5	Mitarbeiterqualifikation	181
7.5.1	Fähigkeit, Modelle lesen zu können	183
7.5.2	Fähigkeit, Modelle erstellen zu können	184
7.5.3	Weitere Fähigkeiten im Zusammenhang mit MBT	185
7.6	Steuerung mit Metriken und Testendekriterien	185
7.6.1	Produktmetriken	186
7.6.2	Projektmetriken	187
7.6.3	Prozessmetriken	189
7.6.4	Anforderungs- und modellbasierte Testendekriterien ...	193
7.7	Zusammenfassung	195

8	Intermezzo 2: CarKonfigurator	197
8.1	Testobjekt und Ziel	197
8.2	Beschreibung der Tests als »Papiermodell«	198
8.2.1	Beschreibung der Testschritte	198
8.2.2	Auswahl der Testdaten	200
8.3	Vom Papiermodell zum UML-Modell	203
8.4	Testfallgenerierung: Algorithmus und Ergebnis	208
8.5	Bewertung	210
8.6	Zusammenfassung	211
9	Modellierung und Modellprüfung	213
9.1	Überblick	213
9.2	Erstellung von Testmodellen	214
9.2.1	Grundlegende Elemente von Testmodellen	214
9.2.2	Modellelemente aus dem CarKonfigurator	216
9.2.3	Stabile Knoten-/Kantenbezeichner	217
9.2.4	Modellierung der Testpriorität	218
9.2.5	Fachlich unerwünschte Pfade	219
9.2.6	Modellierung von Testorakeln	221
9.2.7	Verknüpfung von Testmodellen und zu testenden Anforderungen	223
9.2.8	Exkurs: Testfallableitung aus textuellen Spezifikationen	226
9.2.9	Fazit: Motivation eigenständiger Testmodelle	228
9.3	Modellierung von Testfallspezifikationen	229
9.3.1	Das UML2 Testing Profile	229
9.3.2	FIT und FITnesse	233
9.3.3	TTCN-3	234
9.4	Exkurs: Modellierung von Testumgebungen	237
9.5	Nutzung vorhandener Systemmodelle	240
9.6	Achtung: Fallstricke!	241
9.6.1	Parallele Abläufe oder Alternativen?	241
9.6.2	Hierarchische Verfeinerungen	244
9.7	Übergang von Systemmodellen zu Testmodellen	245
9.7.1	Aus demselben Modell Code und Tests generieren?	246
9.7.2	Grenzen von Systemmodellen	246
9.7.3	Ähneln sich Systemmodelle und Testmodelle?	247
9.7.4	Übergänge vom Systemmodell zum Testmodell	249
9.7.5	Zwei Wege zum konkreten Testfall	252

9.8	Qualitätsmerkmale und Ziele von Modellen	253
9.8.1	Grundlegendes zur Qualität von Modellen	253
9.8.2	Korrektheit (inhaltlich)	254
9.8.3	Korrektheit (formal)	254
9.8.4	Einfachheit	255
9.8.5	Verständlichkeit/Lesbarkeit	255
9.8.6	Angemessenheit	255
9.8.7	Änderbarkeit	255
9.8.8	Vollständigkeit (inhaltlich und formal)	256
9.8.9	Widerspruchsfreiheit (inhaltlich und formal)	256
9.8.10	Prüfbarkeit	256
9.8.11	Werkzeugunterstützung	257
9.8.12	Konsistenz zu anderen Modellen	257
9.9	Prüfung von Modellen	258
9.9.1	Modellierungsrichtlinien als Grundlage	258
9.9.2	Reviews	259
9.9.3	Werkzeuggestützte Prüfungen gegen das Metamodell	259
9.9.4	Model Checker	259
9.9.5	Iteratives Modellieren und Generieren	260
9.9.6	Simulation	261
9.9.7	Geeignete Prüfungen zu Qualitätsmerkmalen	262
9.10	Zusammenfassung	263
10	Testauswahlkriterien und Testfallgenerierung	265
10.1	Vorbemerkung zum Umfang des Kapitels	265
10.2	Einführung in die Generierung	266
10.2.1	Was wird generiert – abstrakte oder konkrete Testfälle?	266
10.2.2	Einsatz struktureller Überdeckungskriterien	269
10.2.3	Das Big Picture der Generierungsverfahren	270
10.3	Generierungsverfahren im Einzelnen	272
10.3.1	Anforderungsbasierte Testgenerierung	272
10.3.2	Modellbezogene Testauswahlkriterien	274
10.3.3	Datenflussbasierte Testauswahlkriterien	277
10.3.4	Zustandsbasierte Testgenerierung	278
10.3.5	Datenüberdeckungskriterien	280
10.3.6	Nutzungs- oder nutzungsmusterbezogene Testauswahlkriterien – »Operational Profiles«	282
10.3.7	UTP-basierte TTCN-3-Generierung	287

10.4	Steuerung der Testfallgenerierung	293
10.4.1	Verknüpfung von Testauswahlkriterien	293
10.4.2	Mengenmäßige Begrenzung der Testfälle	295
10.4.3	Adaptive Testfallgenerierung	297
10.4.4	Negativregeln zum Ausschluss bestimmter Modellteile	298
10.5	Testneugenerierung nach Änderungen	300
10.5.1	Versionsmanagement und MBT	301
10.5.2	Pflegeprozesse nach Modelländerungen	302
10.5.3	Nichtüberschreibung manueller Testarbeiten	302
10.5.4	Exkurs: Adjazenzlisten	304
10.6	Zur Generierung weiterer Testartefakte	305
10.7	Allgemeine MBT-Taxonomie	306
10.8	Bewertung der Testauswahlkriterien	308
10.9	Zusammenfassung	309
11	Realisierung, Durchführung und Auswertung	311
11.1	Manuelle Testdurchführung	311
11.2	Automatisierte Testdurchführung	314
11.2.1	Formen der Testautomatisierung	315
11.2.2	Automatisierungsansätze	320
11.2.3	Durchführungsansätze	322
11.2.4	Verbindung zwischen Modellelementen und Keywords	323
11.2.5	MBT und Keyword-Driven Testing – änderungsstabile Testautomatisierung	325
11.3	Auswertung und Berichterstattung	327
11.3.1	Erzeugung von Traceability-Informationen	327
11.3.2	Nutzung der Traceability zur Fortschrittsbewertung	332
11.3.3	Nutzung der Traceability zur Ergebnisbewertung	334
11.3.4	Strategien zur Reduktion der Anzahl von Abweichungsmeldungen	336
11.4	Zusammenfassung	339
Teil III	MBT im Praxiseinsatz	341
12	Werkzeuge für MBT	343
12.1	Werkzeugeinsatz gemäß der MBT-Definition	343
12.2	MBT im Software Lifecycle	344
12.3	Klassifikationsschemata für MBT-Werkzeuge	346
12.4	Zusammenfassung	352

13	MBT erfolgreich einführen	353
13.1	Hindernisse für die Einführung von MBT	353
13.1.1	Unrealistische oder unklare Ziele	354
13.1.2	Falsche Werkzeugauswahl	354
13.1.3	Probleme von übermorgen lösen wollen	355
13.1.4	Erfolg haben	356
13.2	Erfolgsfaktoren für die Einführung von MBT	356
13.2.1	Ziele klar definieren und messbar machen	356
13.2.2	Akzeptanz schaffen	357
13.3	Die Einführung als Projekt planen und durchführen	358
13.3.1	Analysephase	359
13.3.2	Designphase	362
13.3.3	Evaluierungsphase	363
13.4	Werkzeuge richtig auswählen	363
13.4.1	Integrationsfähigkeit	364
13.4.2	Anpassung an Prozesse und Personal	364
13.4.3	Langfristige Perspektive	365
13.4.4	Umgang mit der Testfallexplosion	366
13.4.5	MBT-Ansatz passend zur Prozessreife definieren	366
13.5	MBT und Prozessreife	367
13.5.1	Reifegradbestimmung des Testprozesses	367
13.5.2	MBTPI – Prozessverbesserung für modellbasiertes Testen	370
13.5.3	Eine Roadmap zur modellbasierten Testreife	372
13.5.4	Skalenbereich zum Reifegrad »Modellorientiertes Testen«	375
13.5.5	Skalenbereich zum Reifegrad »Modellgetriebenes Testen«	377
13.5.6	Skalenbereich zum Reifegrad »Modellzentrisches Testen«	379
13.5.7	Abschließende Bemerkungen zum Reifegradmodell	381
13.6	Zusammenfassung	381
14	Wirtschaftlichkeitsbetrachtung	383
14.1	Übersicht über Kostenfaktoren von MBT	383
14.1.1	Initiale Kosten der Einführung von MBT im Unternehmen	384
14.1.2	Initiale Kosten der Einführung im Projekt	385
14.1.3	Laufende Kosten von MBT im Testprozess	386

14.2	Nutzen von MBT	387
14.2.1	Frühzeitige Anforderungvalidierung und Fehlerfindung	388
14.2.2	Ersparnis bei Erstellung und Pflege der Testfälle	388
14.2.3	Minimierung von Testfallmengen	389
14.2.4	Erhöhung der Testqualität	390
14.3	Kostensparnis durch MBT – Fallbeispiel Trapeze ITS	390
14.3.1	Ausgangssituation	391
14.3.2	Kosten der MBT-Einführung	393
14.3.3	ROI-Betrachtung der MBT-Einführung	395
14.4	Zusammenfassung	402
15	Möglichkeiten und Grenzen von MBT	403
15.1	Was modellbasiertes Testen leisten kann	403
15.2	Was modellbasiertes Testen unter Umständen leisten kann	408
15.3	Was modellbasiertes Testen nicht leisten kann	410
15.4	Was modellbasiertes Testen nicht ersetzen kann	411
15.5	Ausblick – was MBT künftig bringen wird	412
15.6	Zusammenfassung	415
Anhang		417
A	Abkürzungsverzeichnis	419
B	Glossar	421
C	Quellen	431
C.1	Literaturangaben	431
C.2	Webseiten	442
	Index	445

Teil I

MBT – Einstieg und Grundlagen

1 Einleitung

Dieses Kapitel erläutert den Hintergrund, vor dem das Buch geschrieben wurde. Hauptauslöser waren der »Hype« um das Schlagwort »modellbasierter Test« sowie die Vielzahl sehr heterogener existierender Definitionen. Diese wurden mittlerweile auf einen pragmatischen gemeinsamen Nenner gebracht. Darauf aufbauend werden die Ziele des Buches formuliert und der sich daraus ergebende Aufbau sowie der Inhalt der folgenden Kapitel skizziert.

1.1 Model Based Testing – nur ein Hype?

Seit nunmehr über 45 Jahren ist die »Softwarekrise« in mehr oder weniger spürbarer Form immer weiter in den Alltag vorgedrungen. Es vergeht kaum ein Tag, an dem nicht über »Software- oder IT-Fehler« berichtet wird. Vor über 35 Jahren erschien unter dem Titel »The Art of Software Testing« die erste Monografie zum Thema Softwaretest [Myers 1979] die seitdem zigfach fast unverändert nachgedruckt wurde. In jüngerer Zeit ist die Literatur zum Softwaretest förmlich explodiert, und mittlerweile wird bei der überwiegenden Mehrzahl von Softwareprojekten nicht mehr gefragt, ob getestet wird, sondern nur noch was, wie und wie viel. Zur Effizienzsteigerung wird dabei insbesondere auf die automatisierte Durchführung der Tests gesetzt.

Modellbasiertes Testen zielt in erster Linie darauf ab, die Automatisierung auf bislang weniger zugängliche Tätigkeiten in früheren Phasen des Testens auszudehnen. Dort werden nach wie vor aus oft umgangssprachlich formulierten Anforderungen und Systemspezifikationen unter Benutzung mehr oder weniger methodischer Verfahren Testfälle manuell erstellt. Das modellbasierte Testen konzentriert sich auf die Erstellung eines oder mehrerer formaler Modelle, anhand derer z.B. die Testfälle nach vorgegebenen Testkriterien automatisch generiert (und auch ausgeführt) werden können.

Über 35 Jahre »The Art of Software Testing«

Was ist modellbasiertes Testen?

Viel testen = gut testen?

Nun bedeutet »viel testen« bzw. »Testen auf Knopfdruck« im Sinne von vielen automatisch generierten Testfällen nicht automatisch auch »gut testen«. Was aber ist »gutes Testen«? Und wie erreicht man es mit modellbasiertem Testen? Gutes Testen soll einerseits Fehler finden, bevor sie der Kunde findet. Andererseits soll gutes Testen das Vertrauen darin erhöhen, dass das Produkt seinen Einsatzzweck wirklich erfüllt. Für gutes Testen muss also einerseits bekannt sein, wo Fehler am wahrscheinlichsten zu finden sind, und andererseits müssen der Einsatzzweck und die Risiken des Produkts definiert sein.

*Effektiv und effizient
Testen!*

Die Fragen nach der Effektivität, also den »richtigen« bzw. wichtigen Tests, sowie der Effizienz, also der »richtigen« bzw. Ressourcen sparenden Erstellung und Durchführung dieser Tests, rücken dabei immer mehr in den Vordergrund. Naturgemäß fokussiert das Management eher auf die Effizienz und damit auf die Automatisierung sich oft wiederholender und schematisch ausführbarer Tätigkeiten. Vor diesem Hintergrund ist der teilweise mit völlig überzogenen Erwartungen an die zu erzielende Produktqualität einhergehende »Hype« der automatischen Testausführungswerkzeuge in den späten 90er-Jahren zu verstehen.

MBT in der Literatur

Mark Utting und Bruno Leguard setzten in einem der ersten Bücher zu MBT [Utting 2007] den Fokus auf die generativen Aspekte des modellbasierten Testens. Der Titel ihres Buches »Practical model-based Testing« legt die Intention der beiden Autoren nahe, das modellbasierte Testen im praktischen, also vor allem im industriellen Einsatz zu beleuchten. Aber wie reif war das modellbasierte Testen für solche Praxiseinsätze tatsächlich?

Sehr interessante und umfassende Informationen hierzu finden sich in einer der ersten Literaturstudien zu MBT [Dias-Neto 2009]. Die Autoren dieser Studie führten eine systematische Klassifikation von insgesamt 406 bis Mitte 2007 erschienenen Publikationen zum modellbasierten Testen durch. 202 davon erwiesen sich als »gehaltvoll« (d.h., es handelte sich nicht um irrelevante oder duplizierte Informationen). Anschließend wurden 85 Publikationen ausgewählt, die sich entweder auf die Modellierung mit der UML oder auf andere aktuelle (d.h. ab 2004 vorgestellte) oder offenkundig interessante (nämlich mindestens in drei anderen Publikationen referenzierte) Modellierungstechniken bezogen. Diese 85 Veröffentlichungen wurden mit folgendem Ergebnis bezüglich ihrer Praxisrelevanz beurteilt:

Praxisbezug	Anzahl gesichteter Veröffentlichungen			
	UML-basiert	Ohne UML	Gesamt	Anteil in %
Spekulation – Beschreibung des Ansatzes ohne Beispiel oder Referenz auf einen realen Einsatz	17	6	23	27
Beispiel – Erläuterung des Ansatzes mit einem Beispiel, aber ohne Beurteilung der Effektivität und Effizienz	22	16	38	45
Machbarkeitsstudie – Bericht über Anwendung des Ansatzes auf ein »Spieldsystem« mit Nachweis der grundsätzlichen Anwendbarkeit	5	8	13	15
Anwendungsbericht – Bericht über realen Einsatz in einem Industrieprojekt und subjektive Bewertung des Erfolgs	0	4	4	5
Evaluierung – Fallstudie in kontrollierter Praxisumgebung mit quantitativer Analyse von Effektivität und Effizienz	3	4	7	8

Tab. 1-1

Ergebnisse der Studie zum modellbasierten Testen [Dias-Neto 2009]

Die beiden letzten Kategorien der Berichte über praktische Anwendungen umfassen lediglich 11 Artikel, also nur 13 % aller vorgestellten Ansätze.

Ist das modellbasierte Testen auch heute nur ein »Hype« oder ist es mittlerweile eine ernst zu nehmende Technologie? Die Analysten und Berater der Gartner Group [URL: Gartner] veröffentlichen jährlich Berichte zu verschiedenen technologiebezogenen Themen, die sogenannten »Hype Cycles« [URL: WikipediaHypeCycle]. Im Hype Cycle des Jahres 2007 zum Thema »Application Development« fand sich das modellbasierte Testen unter dem Begriff »Scriptless Testing« im Bereich des »peak of inflated expectations«, also auf dem »Gipfel der überzogenen Erwartungen«.

Im Jahr 2009 ordnete Gartner das »Scriptless Testing« sogar noch weiter vorne im Hype Cycle an. 2011 war es dann wieder auf dem Gipfel der (überzogenen) Erwartungen, 2011 und 2012 fand es sich auf dem Weg in das Tal der Desillusionierung. Ab 2013 taucht »Scriptless Testing« dann gar nicht mehr im Hype Cycle auf. Das lässt nach der Systematik von Gartner zwei unterschiedliche Schlussfolgerungen zu: Entweder ist die Technologie in Vergessenheit geraten oder sie ist auf dem Weg zur produktiven Nutzung.

Erst wenige Veröffentlichungen über praktische Anwendungen von MBT

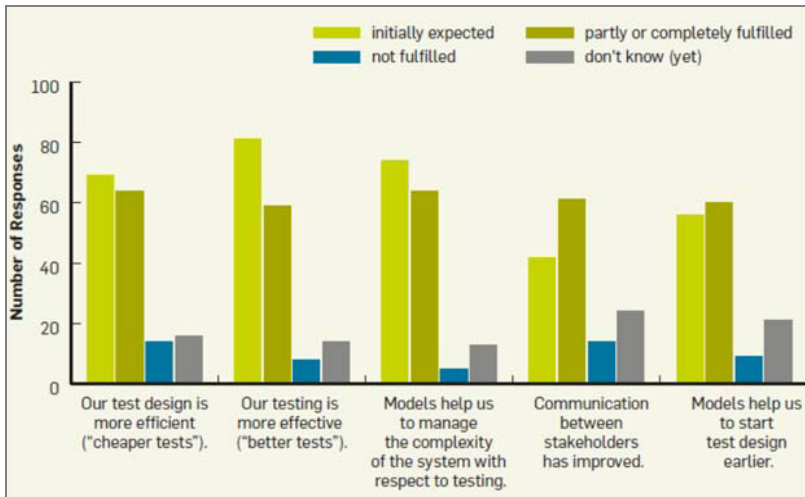
MBT – jahrelang eine Hype-Technologie

Letzteres sehen Mlynarski et al. in ihrem Übersichtsbeitrag zu MBT und folgern:

Model-based testing has been and is applied in many industry projects. [...] there exist a strong literature and empirical body of knowledge. Based on our industry experience, an even wider application of MBT in the industry will be the case. The methods and tools used for MBT gain a great momentum, which makes this technique promising for many companies. [Mlynarski 2012]

Abb. 1-1

Erfüllte Erwartungen in der MBT-Umfrage 2015
(aus: [Binder 2015])



MBT heute

Im Jahr 2014 wollten es Robert Binder et al. genau wissen und führten erneut eine groß angelegte Umfrage zu MBT durch [Binder 2015]. Über hundert MBT-Experten aus aller Welt gaben Auskunft. Binder et al. fanden heraus, dass MBT durchaus die Effektivität und Effizienz des Testens steigert, aber die teilweise überzogenen Erwartungen der Anwender immer noch nicht ganz erfüllen kann (s. Abb. 1-1). Die größten Nutzen werden bei der Testüberdeckung, der Komplexitätsbeherrschung, der automatischen Testfallgenerierung und der Wiederverwendung von Modellen und Modellelementen gesehen, die höchsten Hürden bei der Werkzeugunterstützung, den erforderlichen Kompetenzen und dem Widerstand gegen Veränderungen.

Interessante Einzelaussagen sind, dass MBT-Modelle teilweise lediglich zur Dokumentation genutzt werden, vorhandene Werkzeuge als nutzungsunfreundlich gelten, MBT in einem globalen Unternehmen keine Akzeptanz findet, in einem anderen Unternehmen aber als die einzige praktikable Möglichkeit zum Zuverlässigkeits-Assessment angesehen wird ([URL: MBTSurvey1], [URL: MBTSurvey2], [Binder 2015]).

Welche Erwartungen darf man bei diesen Ergebnissen also tatsächlich bezüglich Effektivität und Effizienz an modellbasierte Vorgehensweisen und Testwerkzeuge haben und wo sind die Grenzen? Wie motiviert man den Einsatz und wie trifft man eine fundierte Auswahl?

Bevor dieses Buch Antworten auf diese und weitere Fragen geben kann, gilt es zunächst zu klären, was genau unter dem Begriff »modellbasierter Test« zu verstehen ist und auf welche Arten von Modellen er sich bezieht.

1.2 Viele Definitionen für modellbasiertes Testen

Bereits 1990 gab Boris Beizer folgende Charakterisierung des Softwaretests:

Testing is a process in which we create mental models of the environment, the program, human nature, and the tests themselves. [...] The art of testing consists of creating, selecting, exploring, and revising models.

[Beizer 1990]

Ausgehend von dieser allgemeinen Definition hat der Testspezialist Robert Binder wohl recht, wenn er fast 10 Jahre später schreibt, Testen wäre immer modellbasiert [Binder 1999]. Tester nutzen nämlich seit jeher vereinfachende Bilder der zu testenden Systeme, um Testfälle und Testszenarien zu definieren. Oftmals sind diese Bilder aber nur als mentale Modelle implizit in ihren Köpfen vorhanden.

Testen ist immer modellbasiert!

Ein wesentliches Ziel von MBT liegt darin, diese Modelle explizit und formal (also eindeutig interpretierbar und maschinenlesbar) zu machen. Dadurch werden komplexe Sachverhalte verständlich, eindeutig und nutzbar für die Automatisierung von Testaktivitäten. Robert Binder folgert:

Ziel von MBT: informelle (mentale) Modelle explizit und formal machen

Test design must be based on both general and specific models.

[Binder 1999]

Im Eintrag »Model-Based Software Testing« der »Encyclopedia of Software Engineering« geben Whittacker und El-Far folgende allgemeine Definition für MBT:

MBT-Definitionen in der Literatur

Model-Based Testing (MBT) is [...] an approach that bases common testing tasks such as test case generation and test result evaluation on a model of the application under test.

[Marciniak 2002]

Die allgegenwärtige Wikipedia enthält folgende Erläuterung:

Modellbasiertes Testen (MBT) ist ein Oberbegriff für die Nutzung von Modellen zur Automatisierung von Testaktivitäten [bzw. zur] Generierung von Testartefakten im Testprozess. Darunter fällt insbesondere die Generierung von Testfällen aus Modellen (z.B. unter Verwendung der UML), die das Sollverhalten des zu testenden Systems beschreiben.

[URL: Wikipedia]

Dort finden sich auch Erläuterungen für die ähnlich gelagerten Begriffe »modellzentriertes Testen« und »modellgetriebene Testentwicklung«.

In der oben skizzierten Literaturstudie zu MBT wird modellbasiertes Testen wie folgt abgegrenzt:

MBT approaches help automatically generate test cases using models extracted from software artifacts.

[Dias-Neto 2009]

Conrad interpretiert den Begriff dagegen in Richtung des Testens von Modellen und verknüpft ihn eng mit der modellbasierten Entwicklung:

Im Hinblick auf den Softwaretest eröffnet die Modellbasierte Entwicklung [...] die Möglichkeit, schon das ausführbare Modell des Systems einer analytischen Qualitätssicherung zu unterziehen (Modelltest) [...] Ein derartiger den Modellbasierten Entwicklungsprozess begleitender Test wird als Modellbasierter Test bezeichnet.

[Conrad 2004]

In einem Artikel von Holger Schlingloff liest sich das so:

Beim modellbasierten Testen geht es dabei sowohl um den Test von Dokumenten einer modellbasierten Entwicklung wie zum Beispiel Systemmodellen, Implementierungsmodellen oder generiertem Code als auch um die Ableitung von Testsuiten aus Modellen (Testfallgenerierung).

[Schlingloff 2007]

Mark Utting und Bruno Legeard legen in ihrem oben bereits erwähnten Buch »Practical Model-Based Testing« den Fokus auf die generativen Aspekte des modellbasierten Testens und schreiben:

The following are the four main approaches known as model-based testing:

1. *Generation of test input data from a domain model.*
2. *Generation of test cases from an environment model.*
3. *Generation of test cases with oracles from a behavior model.*
4. *Generation of test scripts from Abstract tests.*

[Utting 2007]

Die Erfinder des UML2 Testing Profile (UTP, s. Abschnitt 9.3.1) fügen diesen Definitionen als weitere Fassetten die formale Beschreibung bzw. Modellierung der Tests selbst zu:

Model-based testing requires the systematic and possibly automatic derivation of tests from models. In our case, UML is the language for specifying models and UTP the formalism to describe the derived tests.

[Baker 2008]

Im Juli 2011 erschien mit der ETSI-Spezifikation ES 202 951 [ETSI 2011] erstmals eine international gültige Spezifikation u.a. zum Thema MBT, die MBT sehr allgemein definiert als

An umbrella of approaches that generate tests from models.

[ETSI 2011]

Welche grundsätzlichen gemeinsamen Merkmale von MBT lassen sich nun in all diesen Definitionen bzw. den sie enthaltenden Publikationen finden?

Beim modellbasierten Testen wird die zu testende Software, ihre Umgebung oder der Test selbst über Modelle betrachtet, die sich auf besondere Eigenschaften des Testobjekts bzw. der Tests konzentrieren, meist auf deren Verhalten. Diese Modelle können eigenständig, parallel zu Entwicklungsmodellen erstellt oder aber aus diesen abgeleitet werden. Wie alle softwaretechnischen Artefakte sind sie Gegenstand der Qualitätssicherung. Aus den Modellen werden – mit teilweise seit vielen Jahren bekannten Verfahren – u.a. Testfälle automatisch generiert, hinsichtlich der Testdaten sowie technologisch konkretisiert und über entsprechende Testrahmen oder -adapter ausgeführt.

Merkmale von MBT

Diese Erkenntnisse führen zu einer vorläufigen pragmatischen Definition von MBT, die in Kapitel 4 im Detail betrachtet wird:

Modellbasiertes Testen umfasst mindestens einen der beiden folgenden Aspekte:

- Tests modellieren
- Tests aus Modellen generieren

(Vorläufige)

Definition

von MBT

Nach diesen grundlegenden Begriffsklärungen können nun die Ziele des Buches formuliert werden.

1.3 Ziele des Buches

Das globale Ziel des Buches ist die Beantwortung der Frage, ob sich der Einsatz von MBT in Ihrem Hause lohnt und wenn ja, welche der vielen in der Literatur vorgeschlagenen Methoden und Techniken sowie welche der verfügbaren MBT-Werkzeuge die passenden sein könnten. Hieraus ergeben sich die folgenden Teilziele.

Rahmen und Definitionen zu MBT schaffen

Wichtigstes Ziel des Buches ist – mit Blick auf die oben aufgeführten und durchaus unterschiedlichen Definitionen modellbasierten Testens –, einen belastbaren und möglichst allgemeingültigen Rahmen für das Thema zu schaffen. Dazu gehört zwingend, einige Definitionen einzuführen. Diese werden mit einigen der bereits existierenden Quellen teilweise im Widerspruch stehen oder Unterschiede aufweisen.

Da das Thema MBT selbst noch vergleichsweise jung ist, lässt sich das nicht vermeiden – es existieren eben noch unterschiedliche Ansichten darüber, was unter »modellbasiertem Testen«, »modellgetriebenem Testen« oder aber »modellzentrischem Testen« zu verstehen ist. Wir haben uns im Zuge der Buchentstehung auf ein Begriffsmodell und sogar auf ein MBT-Reifegradmodell geeinigt, das wir für tragfähig halten. Hier mögen Sie selbst entscheiden, ob Sie einige der hier getroffenen Festlegungen für ungeeignet halten – in solchen Fällen würden wir uns sehr freuen, wenn Sie uns davon in Kenntnis setzen.

Rüstzeug für weiterführende Fachliteratur vermitteln

Der so geschaffene Rahmen soll Sie insbesondere in die Lage versetzen, die stetig anwachsende Fachliteratur zum Thema MBT (siehe z.B. das Literaturverzeichnis für eine aktuelle Auswahl der wichtigsten Beiträge) einerseits zu verstehen, andererseits einordnen und bewerten zu können. Immer noch ist zuweilen die eine oder andere unreife oder vielleicht auch vom Marketing getriebene Aussage anzutreffen. Das hier vermittelte Wissen soll es ermöglichen, die Spreu vom Weizen zu trennen.

MBT in der Breite, aber nicht immer in der Tiefe präsentieren

Trotz allem trägt das Buch nicht umsonst den Titel »Basiswissen«. Es hat ausdrücklich nicht das Ziel, jedes hier genannte Thema und jeden möglichen Übergang von Modellen zu Testartefakten in gleicher Tiefe und Gründlichkeit zu präsentieren. Dies liegt u.a. daran, dass die UML (Unified Modeling Language) der OMG in der Welt der vielen in der IT verwendeten Modellierungstechniken und -sprachen unbestreitbar eine führende Rolle einnimmt.

Folglich steht der Übergang von UML-Modellen hin zu Testfällen deutlich im Vordergrund. Es würde den Rahmen eines »Basiswissen«-Buches sprengen, in gleichem Umfang etwa BPMN, EPK-, SA/SD-Modelle oder Petri-Netze sowie die Techniken zum Model Checking im Hinblick auf modellbasierte Testmethoden zu beleuchten. Hier verweisen wir jeweils auf weiterführende Fachliteratur.

Unterschiede zum konventionellen Testprozess klären

Aus mehreren guten Gründen setzen wir in unserer Darstellung auf den Lehrplänen des ISTQB® und damit auch auf den in derselben Buchreihe wie das vorliegende Buch erschienenen Begleitbüchern zum Certified-Tester-Ausbildungsschema auf. Darin wird der sogenannte fundamentale Testprozess präsentiert, der aus insgesamt fünf Phasen mit zugehörigen Rollen und typischen (Test-)Artefakten besteht.

Eine der Kernfragen, denen sich dieses Buch widmet, lautet, ob sich dieser Testprozess beim modellbasierten Testen verändert und wenn ja, wie ein MBT-Testprozess im Vergleich dazu aussieht – wozu dann auch entsprechende Rollen und Artefakte gehören.

Unrealistische Erwartungen an MBT beseitigen

MDA- und MDSD-Ansätze (*Model-Driven Architecture, Model-Driven Software Development*) haben in den letzten Jahren das Bild gefestigt, dass es unter geeigneten Voraussetzungen möglich ist, aus Modellen Code oder wenigstens Codeteile zu generieren und damit einen Effizienz- und Qualitätsgewinn zu erzielen. Vor diesem Hintergrund verwundert es nicht, wenn regelmäßig die Frage gestellt wird: »Wenn wir aus unseren Modellen so gut Code generieren können – können wir dann nicht auch gleich noch unsere Tests aus ihnen generieren?«

Das Buch wird klarmachen, dass dies zu kurz gedacht ist. Nicht jedes Modell eignet sich gleich gut für Generierungen in Richtung Test, und die Formulierung »Tests generieren« ist zu undifferenziert: Was genau wird generiert? Testdaten? Testfälle – abstrakte oder konkrete? Testskripte? Testergebnisse? Testrahmen? Insbesondere wird klar werden, dass es »das MBT«, d.h. modellbasiertes Testen »von der Stange«, nicht gibt.

Nutzen und Potenziale von MBT identifizieren

Stattdessen ist es Aufgabe des Buches, deutlich herauszustellen, was MBT leisten kann und was dazu benötigt wird. Es wird sich zeigen, dass MBT eine sinnvolle Ergänzung zu den bekannten »konventionellen« Testansätzen darstellt. Die konkrete Ausgestaltung und Einführung sind aber in jedem Projekt individuell vorzunehmen. Insbeson-

dere soll deutlich werden, in welchen Situationen MBT helfen kann, Probleme im Testvorgehen zu lösen, und für welche Probleme dies möglich ist.

Eignungsprüfung für eigene Testprojekte ermöglichen

Ebenfalls wird klar werden, dass modellbasiertes Testen – wie jede andere Test- bzw. Qualitätssicherungsmaßnahme auch – mit Aufwand und somit mit Investitionen verbunden ist. Außerdem existiert eine Reihe von Faktoren, die für einen erfolgreichen MBT-Einsatz erfüllt sein müssen. Beide Punkte müssen im Vorfeld geprüft werden, wenn man sich über den Einsatz vom MBT im eigenen Projekt Gedanken macht: Sind die Rahmenbedingungen dafür gegeben? Und ist auf einen messbaren Mehrwert bzw. ROI (Return on Investment) zu hoffen, wenn MBT im Projekt eingeführt wird?

Auch zu diesen beiden wichtigen Fragen will das Buch Hilfestellung und Hinweise geben – bis hin zu einem Fragenkatalog zur Klärung der Voraussetzungen für den erfolgreichen MBT-Einsatz.

Bei der MBT-Werkzeugauswahl unterstützen

Prozesse und Werkzeuge gehen Hand in Hand, und so verwundert es nicht, dass modellbasiertes Testen teilweise auch den Einsatz neuer bzw. neuartiger Testwerkzeuge mit sich bringt. Auch hier gilt, dass es einerseits große Unterschiede zwischen den aktuell vorhandenen Werkzeugen am Markt gibt und dass andererseits nur mit der Anschaffung eines Werkzeugs nicht bereits alles getan ist. Das Buch führt verschiedene MBT-Werkzeugkategorien ein, stellt einige Vertreter kurz vor und hilft bei der Auswahl passend zur gewählten MBT-Strategie.

An Beispielen veranschaulichen

Wo möglich, werden praxisnahe Beispiele herangezogen, um Aussagen zu MBT zu illustrieren. Dabei haben wir versucht, mit zwei durchgängigen, leicht nachvollziehbaren Beispielen möglichst viele Aspekte von MBT zu erläutern.

1.4 Aufbau des Buches

Nach dieser kurzen Einführung in das modellbasierte Testen möchten wir Ihnen noch einen Überblick darüber geben, was Sie in den folgenden Kapiteln erwartet.

Teil I (MBT – Einstieg und Grundlagen) ist optional je nach Vorkenntnissen zu überspringen und besteht aus dieser Einleitung sowie den Kapiteln 2 bis 5. Nach Lektüre dieses Teils sollten Sie in der Lage sein, einer Diskussion, einem Fachartikel oder -buch zu MBT (und insbesondere den restlichen Kapiteln dieses Buches) folgen zu können.

- Kapitel 2 erläutert in knapper Form die wichtigsten in diesem Buch verwendeten Grundbegriffe des Testens und den fundamentalen Testprozess nach ISTQB®. Am Ende des Kapitels werden grundlegende Herausforderungen genannt, denen sich das Testen heute gegenüber sieht und die durch moderne Technologien wie MBT adressiert werden können.
- Kapitel 3 führt Grundlagen der Modellierung ein. Der Schwerpunkt liegt hierbei auf der Syntax und Semantik der UML.
- In Kapitel 4 wird die Definition für »modellbasiertes Testen« erläutert, die wir dem Buch zugrunde gelegt haben. Davon abgeleitet ist eine Darstellung der wesentlichen Modellkategorien und Einsatzszenarien für MBT.
- Kapitel 5 ist ein frühes Intermezzo zwischen den methodischen Teilen des Buches. Es detailliert anhand des Fallbeispiels zum modellbasierten Test einer elektronischen Türsteuerung ein typisches Einsatzszenario von MBT.

Teil II (MBT im Testprozess), ist das »Herz« dieses Buches, besteht aus den Kapiteln 6 bis 11 und stellt MBT als Prozess schrittweise dar.

- In Kapitel 6 wird zunächst der Rahmen gebildet, indem erläutert wird, wie der Testprozess unter Einbeziehung von MBT zu gestalten ist. Des Weiteren stellt das Kapitel dar, wie dieser Prozess für unterschiedliche Teststufen anzupassen ist und wie er mit unterschiedlichen Vorgehensmodellen zur Softwareentwicklung zusammenwirkt. Die Betrachtungen dieses Kapitels sind unabhängig von der konkreten Modellierungsmethode.
- Kapitel 7 konkretisiert die Aktivitäten zur Planung und Steuerung des Testprozesses mit Hinblick auf die Nutzung von MBT. Dabei wird zwar eine iterativ-inkrementelle Vorgehensweise angenommen, aber – wie schon im vorangegangenen Kapitel – keine spezifische Modellierungsmethode vorausgesetzt.
- Kapitel 8 ist als weiteres Intermezzo und als Klammer für die beiden darauf folgenden Kapitel zu betrachten. Anhand der hypothe-

tischen Anwendung »CarKonfigurator«, die als Testbeispiel weithin bekannt ist, werden erste Zusammenhänge zwischen Testmodellen und Generierungsverfahren vermittelt.

- Kapitel 9 greift dieses Beispiel auf und beschreibt die Testanalyse und den Testentwurf auf Basis von MBT. Es diskutiert einige industriell einsetzbare Modellierungsmethoden auf der Basis von System- und Testmodellen und deren Zusammenspiel mit Testfallgeneratoren. Darüber hinaus erläutert es einige »Fallstricke« der Modellierung sowie die wesentlichen Maßnahmen zur Qualitätssicherung von Modellen.
- Im Fokus des Kapitels 10 steht die Testrealisierung aus Sicht von MBT. Es wird erläutert, wie Testfallgeneratoren funktionieren und was das Ergebnis des Generiervorgangs ist. Generierungsalgorithmen, Abdeckungskriterien und Methoden zur Steuerung von Generatoren werden diskutiert.
- Kapitel 11 schließt die Betrachtung der Phasen des fundamentalen Testprozesses ab und beleuchtet die Testdurchführung und die Bewertung des Tests. Dabei wird die Brücke sowohl zur manuellen als auch zur automatisierten Durchführung generierter Testfälle geschlagen. Die Bewertung der Endkriterien und die Berichterstattung der Testergebnisse auf Basis von Modellinformationen stellen den zweiten Schwerpunkt des Kapitels dar.

In Teil III (MBT im Praxiseinsatz) werden aus den Kernaussagen des zweiten Teils weiterführende Themen entwickelt, um den Leserinnen und Lesern dabei zu helfen, MBT im eigenen Projekt einzusetzen. Dieser Teil umfasst die Kapitel 12 bis 15.

- Kapitel 12 stellt eine Systematik zur Kategorisierung von Softwarewerkzeugen im MBT-Umfeld bereit. Ausgehend von diesen Grundlagen können MBT-Werkzeuge evaluiert und auf die Einsetzbarkeit im eigenen Umfeld hin bewertet werden.
- Kapitel 13 widmet sich der Frage, wie MBT in einem konkreten Projekt oder einer Organisation systematisch und erfolgreich eingeführt wird. Die Grundlage hierfür bildet ein Reifegradmodell für modellbasiertes Testen.
- Kapitel 14 gibt dem Leser mittels einer Fallstudie einen Leitfaden an die Hand, mit dem die Wirtschaftlichkeit von MBT bewertet werden kann.
- Kapitel 15 schließlich zieht ein Fazit aus Sicht der Autoren, welches Potenzial MBT heute im industriellen Einsatz erschließen kann. Es werden aber auch die Grenzen der im Buch aufgegriffenen Ansätze zusammengefasst. Zum Abschluss wagen wir einen Blick in die (nähere) Zukunft von MBT.

Jedes Kapitel (mit Ausnahme dieser Einleitung) endet mit einer Zusammenfassung, in der die wichtigsten Punkte kurz und prägnant dargestellt werden. Einsteigern in die Thematik MBT, die sich nur einen Erstüberblick verschaffen wollen und die kein entsprechendes Training absolvieren möchten, empfehlen wir mit den Kapiteln bzw. Abschnitten 2, 3.1/3.2 sowie 4 zu starten. Die Abschnitte 6.1 und 6.2 würden als prozessuales Grundwissen diesen Erstüberblick bereits schon wieder abschließen. Für den »gestressten Manager« bzw. Teamleiter oder für andere Entscheider sind die Kapitel 13 bzw. 14 zu empfehlen, da darin sowohl die wirtschaftlichen ROI-Faktoren als auch die Faktoren zur erfolgreichen Einführung von MBT behandelt werden. Für diejenigen, die sich gezielt auf die Prüfung zum ISTQB® Certified Model-Based Tester vorbereiten wollen, ist als Direkteinstieg Abschnitt 1.5 erstellt worden. Ansonsten darf man unser Buch auch ganz normal von vorne bis hinten durchlesen.

1.5 Leseanleitung für die Qualifikation zum ISTQB® Certified Model-Based Tester

Der Hauptteil des Buches orientiert sich grob an den Phasen bzw. Aktivitäten des ISTQB®-Testprozesses. Dies wurde im Lehrplan zum ISTQB® Certified Model-Based Tester nur teilweise übernommen. Die folgende Leseanleitung dient daher zur zielgerichteten Vorbereitung auf die Zertifizierungsprüfung zum ISTQB® Certified Model-Based Tester, für die wir Ihnen viel Erfolg wünschen!

*MBT – seit 2015 im ISTQB®
Certified-Tester-Schema*

Einsteigern in die Materie legen wir ein entsprechendes Training bei einem vom GTB e.V. akkreditierten Trainingsunternehmen ans Herz [URL: GTB]. Vielleicht haben Sie dieses Buch ja sogar als Begleitmaterial zu einem solchen Training erhalten! In der untenstehenden Tabelle finden Sie Einstiegspunkte in die Kapitel bzw. Abschnitte des Buches bezüglich der Inhalte des Lehrplans.

Einsteigern, die kein entsprechendes Training absolvieren möchten, empfehlen wir, das Buch zunächst von vorne bis hinten durchzulesen. Studieren Sie danach den Lehrplan, und konsultieren Sie dabei noch einmal vertiefend die Kapitel bzw. Abschnitte des Buches gemäß der Tabelle.

Fortgeschrittene MBT-Anwender können nach einem ersten »Querlesen« des Buches beim Studium des Lehrplans direkt die Kapitel bzw. Abschnitte des Buches gemäß Tabelle 1–2 konsultieren.

Essenzielle Buchkapitel für die Zertifizierungsprüfung

Insgesamt decken folgende Buchkapitel den ISTQB®-Lehrplan ab:

- Kapitel 3, Abschnitte 3.1, 3.2, 3.5, 3.6
- Kapitel 4, Abschnitte 4.3, 4.4
- Kapitel 6, Abschnitte 6.1, 6.2, 6.3, 6.5
- Kapitel 7, Abschnitte 7.2, 7.3, 7.6
- Kapitel 9, Abschnitte 9.2, 9.5, 9.6, 9.7, 9.8, 9.9
- Kapitel 10, Abschnitte 10.2, 10.3, 10.4, 10.5, 10.8
- Kapitel 11, Abschnitte 11.1, 11.2, 11.3
- Kapitel 12
- Kapitel 13, Abschnitte 13.2, 13.3, 13.4, 13.5
- Kapitel 14, Abschnitt 14.1
- Kapitel 15, Abschnitte 15.1, 15.2, 15.3, 15.4
- sowie die beiden MBT-Beispiele in den Kapiteln 5 und 8

Tab. 1-2
Einstiegspunkte gemäß
ISTQB®-MBT-Lehrplan

Lehrplan	Buch	Seiten
1 Introduction to Model-Based Testing	1 Einleitung	3
1.1 Objectives and Motivations for MBT	2.3 Herausforderungen beim Testen heute	38
1.1.1 Main Motivations for MBT	15.1 Was modellbasiertes Testen leisten kann	403
	15.2 Was modellbasiertes Testen unter Umständen leisten kann	408
1.1.2 Misleading Expectations and Pitfalls of MBT	15.3 Was modellbasiertes Testen nicht leisten kann	410
	15.4 Was modellbasiertes Testen nicht ersetzen kann	411
1.2 MBT Activities and Artifacts in the Fundamental Test Process	6 MBT und der Testprozess	135
1.2.1 MBT Specific Activities 1.2.2 Essential MBT Artifacts (Inputs and Outputs)	6.1 Ausprägungen von MBT im Testprozess	136
	6.2 Allgemeine Integration von MBT in den Testprozess	140
	6.3 Abhängigkeit des MBT-Prozesses von den verwendeten Modellkategorien	144
	7.3 Aktivitätenplanung und Aufwandsschätzung	174



Lehrplan		Buch		Seiten
1.3	Integrating MBT into the Software Development Lifecycles	6.5	MBT und der Entwicklungsprozess	159
1.3.1	MBT in Sequential and Iterative Software Development Lifecycles	6.5.2	Betrachtung verschiedener Vorgehensmodelle im Zusammenhang mit MBT	160
1.3.2	Supporting Requirements Engineering	3.2	Allgemeiner Vergleich von Texten und Modellen	50
		4.3	MBT im Projektkontext	107
2	MBT Modeling	3	Einführung in die Modellierung	47
2.1	MBT Modeling			
2.1.1	MBT Modeling Activities	4	MBT – ein Einstieg	105
2.1.2	Subject and Focus of MBT Models	3.1	Wat is'n Modell?	47
		4.4	MBT-Modellkategorien	109
2.1.3	MBT Models Depend on Test Objectives	7.2	Bestimmung der Teststrategie	172
2.2	Languages for MBT Models	3.1	Wat is'n Modell?	47
2.2.1	Main Categories of Modeling Languages for MBT	3.5	Modellierung statischer Strukturen	58
2.2.2	Language Categories Relevant for Different Systems and Project Objectives	3.6	Modellierung dynamischen Verhaltens	69
		4.3	MBT im Projektkontext	107
2.3	Good Practices for MBT Modeling Activities	9	Modellierung und Modellprüfung	213
2.3.1	Quality Characteristics for MBT Models	9.8	Qualitätsmerkmale und Ziele von Modellen	253
2.3.2	Typical Mistakes and Pitfalls in MBT Model Design	9.6	Achtung: Fallstricke!	241
2.3.3	Linking Requirements and Process Related Information to the MBT Model	9.2.7	Verknüpfung von Testmodellen und zu testenden Anforderungen	223
2.3.4	Modeling Guidelines for MBT	9.9.1	Modellierungsrichtlinien als Grundlage	258
2.3.5	Reuse of Existing System Design or Requirements Models	6.3	Abhängigkeit des MBT-Prozesses von den verwendeten Modellkategorien	144
		9.5	Nutzung vorhandener Systemmodelle	240
2.3.6	Tool Support for Modeling Activities	9.8.11	Werkzeugunterstützung	257
		12	Werkzeuge für MBT	343
2.3.7	Iterative Model Development, Review and Validation	9.9	Prüfung von Modellen	258

Lehrplan	Buch	Seiten
3 Selection Criteria for Test Case Generation	10 Testauswahlkriterien und Testfallgenerierung	265
3.1 Classification of MBT Test Selection Criteria	10.2 Einführung in die Generierung	266
3.1.1 Test Selection Criteria	10.3 Generierungsverfahren im Einzelnen	272
3.1.2 Test Case Selection in Practice	10.4 Steuerung der Testfallgenerierung	293
3.1.3 Examples of Test Selection Criteria	10.3 Generierungsverfahren im Einzelnen	272
	5 Intermezzo 1: Türsteuerung	119
	8 Intermezzo 2: CarKonfigurator	197
3.1.4 Relation to Foundation Level Test Design Techniques	8.2.2 Auswahl der Testdaten	200
	10.3.1 Anforderungsbasierte Testgenerierung	272
	10.3.5 Datenüberdeckungskriterien	280
3.2 Applying Test Selection Criteria	6.1 Ausprägungen von MBT im Testprozess	136
3.2.1 Degree of Automation in Test Generation		
3.2.2 Pros and Cons of Specific Test Selection Criteria	10.8 Bewertung der Testauswahlkriterien	308
3.2.3 Good Practices of MBT Test Selection	10.4.1 Verknüpfung von Testauswahlkriterien	293
4 MBT Test Implementation and Execution	11 Realisierung, Durchführung und Auswertung	311
4.1 Specifics of MBT Test Implementation and Execution	9.7 Übergang von Systemmodellen zu Testmodellen	245
4.1.1 Abstract and Concrete Test Cases in the MBT Context	10.2.1 Was wird generiert – abstrakte oder konkrete Testfälle?	266
4.1.2 Different Kinds of Test Execution	11.1 Manuelle Testdurchführung	311
	11.2 Automatisierte Testdurchführung	314
4.1.3 The Impact of Changes on the MBT Artifacts	10.5 Testneugenerierung nach Änderungen	300
4.2 Activities of Test Adaption in MBT	11.2 Automatisierte Testdurchführung	314

Lehrplan	Buch	Seiten
5 Evaluating and Deploying an MBT Approach	13 MBT erfolgreich einführen	353
5.1 Evaluate an MBT Deployment	13.5 MBT und Prozessreife	367
5.1.1 ROI Factors for MBT Introduction	14 Wirtschaftlichkeitsbetrachtung	383
5.1.2 Organizational Objectives and their Relationship to the Characteristics of the MBT Approach	13.5.2 MBTPI – Prozessverbesserung für modellbasiertes Testen	370
5.1.3 Metrics and Key Performance Indicators	11.3 Auswertung und Berichterstattung	327
	10.3 Generierungsverfahren im Einzelnen	272
	13.2.1 Ziele klar definieren und messbar machen	356
	7.6 Steuerung mit Metriken und Testendekriterien	185
5.2 Manage and Monitor the Deployment of an MBT Approach	13 MBT erfolgreich einführen	353
5.2.1 Good Practices when Deploying MBT	13.3 Die Einführung als Projekt planen und durchführen	358
5.2.2 Cost Factors of MBT	14.1 Übersicht über Kostenfaktoren von MBT	383
5.2.3 Integration of the MBT Tool	13.4 Werkzeuge richtig auswählen	363
9 Appendix A – Simple Modeling Language	3.6 Modellierung dynamischen Verhaltens	69
9.1 A Simple Graphical Modeling Language for Workflows	3.6.1 Aktivitätsmodellierung	70
9.2 A Simple Graphical Modeling Language for State Transition Diagrams	3.6.3 Zustandsmodellierung	76

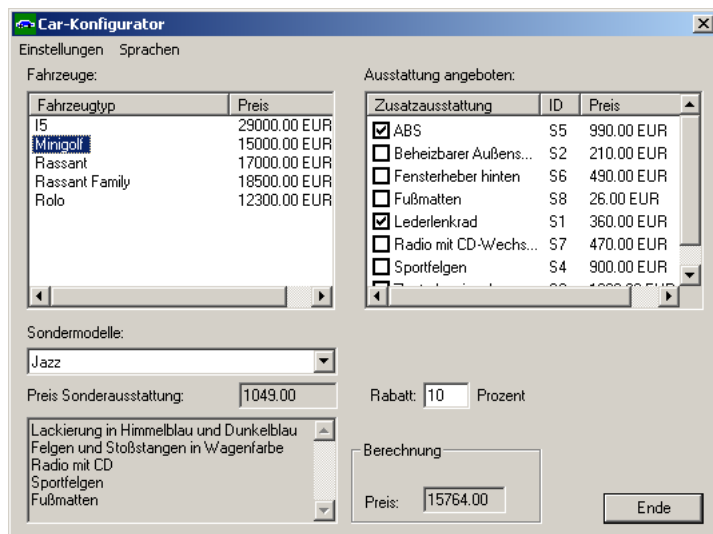
1.6 Zwei Fallbeispiele

Die folgenden beiden einfachen hypothetischen Systeme dienen im gesamten Buch als durchgängige Beispiele für Modellierungs- und Testfallgenerierungstechniken.

1.6.1 Produktskizze CarKonfigurator

Bei dem ersten Fallbeispiel handelt es sich um den aus [Spillner 2012] bekannten CarKonfigurator, der für ein neues elektronisches Verkaufssystem, genannt VirtualShowRoom (VSR), eines Automobilkonzerns entwickelt wird. Diese kleine Applikation ist eine Art Taschenrechner, mit dem Autohändler bzw. deren Kunden den Preis des Wunschautos berechnen können (s. Abb. 1–2). Die Anwendung soll in der Endausbaustufe weltweit bei allen Händlern eines Automobilkonzerns installiert werden.

Abb. 1–2
Benutzerschnittstelle des
CarKonfigurators



Folgende Anforderungen an den CarKonfigurator sind relevant:

Funktionale Anforderungen

*Funktionale
Anforderungen*

Das System ermöglicht die interaktive Konfiguration von Fahrzeugen am Bildschirm (Modellauswahl, Farbe, Ausstattung usw.). Nach der Auswahl eines Fahrzeugtyps zeigt das System mögliche Sondermodelle und zusätzlich angebotene Ausstattungsvarianten an. Zu jeder Auswahl ermittelt das System sofort den jeweiligen Listenpreis.

Während der Auswahl von Fahrzeugtyp sowie – optional – Sondermodell, Zusatzausstattungen und Händlerrabatt wird der Gesamtpreis des so konfigurierten Autos immer sofort aktualisiert und angezeigt.

Das System bietet mehrere Sprachen zur Benutzerführung an, aus der eine ausgewählt werden kann. In der ersten Version sollen dies Deutsch und Englisch sein.

Benutzbarkeitsanforderungen

Benutzer des Systems sind Kunden, die entweder unterstützt durch einen Verkäufer oder aber vollkommen selbstständig ihr Wunschfahrzeug am Bildschirm konfigurieren möchten. Daher muss das System intuitiv und ohne jegliche Einarbeitungszeit benutzbar sein.

Benutzbarkeitsanforderungen

1.6.2 Produktskizze Türsteuerung

Im zweiten Fallbeispiel geht es um die Steuerungssoftware für einen elektronischen Schiebetürantrieb, der für den Einsatz an Außentüren öffentlich zugänglicher Gebäude (z.B. Krankenhäuser, Ämter) konzipiert ist. Während im Fallbeispiel CarKonfigurator neben den funktionalen Anforderungen vor allem die Benutzbarkeit als nicht funktionale Anforderung im Mittelpunkt steht, sind bei der Türsteuerung neben der Funktion hauptsächlich Sicherheitsanforderungen zu beachten.

Ein elektronischer Schiebetürantrieb

Folgende Anforderungen an die Türsteuerung sind dokumentiert:

Funktionale Anforderungen

Der Antrieb öffnet die Tür bei Annäherung von Personen automatisch. Hierzu wird er mit jeweils einem Annäherungssensor auf der Innen- und der Außenseite der Tür verbunden.

Ob die Schiebetür vollständig geöffnet oder geschlossen ist, wird dem Antrieb über zwei Anschlagssensoren mitgeteilt. Außerdem gibt es einen Einklemmsensor, der die Verletzung von Personen vermeiden soll, die sich während des Schließvorgangs im Türrahmen befinden. Spricht während des Schließvorgangs der Annäherungssensor oder der Einklemmsensor an, so stoppt der Schließvorgang und der Motor beginnt die Tür wieder zu öffnen.

*Funktionale Anforderungen:
automatisches Öffnen
und Schließen der Tür*

Ist die Tür geöffnet und nähert sich keine weitere Person, so soll die Tür sich nach einer Wartezeit von 10 Sekunden wieder schließen. Falls während dieser 10 Sekunden eine weitere Person die Tür durchschreitet, beginnt die Wartezeit erneut.

Über eine Netzwerkleitung kann der Antrieb verriegelt werden, wenn sich die Tür im geschlossenen Zustand befindet. Durch diese Leitung wird die zentrale Verriegelung des Gebäudes durch die Nachtwache unterstützt.

Sicherheitsanforderungen

*Sicherheitsanforderungen:
Verhalten der
Türsteuerung im
Gefahrenfall*

Da es sich um die Eingangstür eines öffentlichen Gebäudes handelt, muss die Tür im Gefahrenfall jederzeit manuell von innen entriegelt und dauerhaft geöffnet werden können. In diesem Fall kann eine Person die manuelle Entriegelung betätigen und anschließend die Tür aufschieben; falls zu diesem Zeitpunkt der Antrieb noch funktioniert, soll die Steuerung das Aufschieben durch Anstoßen eines automatischen Öffnungsvorgangs unterstützen und, wenn die Tür vollständig geöffnet ist, den Antrieb bei geöffneter Türposition verriegeln.

2 Testen heute

Dieses Kapitel erläutert zunächst wichtige Grundbegriffe des Softwaretests und wiederholt den sogenannten fundamentalen Testprozess – mit dem Testen vertraute Leser können diese ersten Abschnitte getrost überspringen. Sie dienen einerseits als kurze Wiederholung, andererseits als Startpunkt für die Untersuchung der Besonderheiten des modellbasierten Testens. Um MBT im Verlauf des Buches in seiner Bedeutung besser einordnen zu können, beschäftigt sich das Kapitel zum Schluss mit der Frage, vor welchen Herausforderungen das Softwaretesten heute steht.

2.1 Grundbegriffe von Qualitätssicherung und Testen

In diesem Abschnitt werden in aller Kürze einige grundlegende Begriffe erläutert, die zum Verständnis der weiteren Ausführungen unabdingbar sind. Die Erläuterungen lehnen sich eng an die Literatur zu den Aus- und Weiterbildungsmaßnahmen nach dem »ISTQB® Certified Tester«-Schema [URL: ISTQB] an, das weltweit eine sehr große Akzeptanz und Verbreitung gefunden hat. In Deutschland wird das Schema vom German Testing Board e. V. (GTB) gepflegt und vertreten, auf dessen Webseiten die deutschsprachigen Lehrpläne und Unterlagen abrufbar sind [URL: GTB]. Nähere Ausführungen zu den Grundlagen des Softwaretests finden sich in [Spillner 2012]. Weiterführende Inhalte zum Testmanagement enthält [Spillner 2014], solche zum fachlichen und technischen Testen vermittelt [Bath 2015] oder [Spillner 2016].

Die Erkenntnis, dass Qualität nicht im Nachhinein in ein Produkt »hineingepüffelt« werden kann, hat zur Unterscheidung der konstruktiven und analytischen Maßnahmen der Qualitätssicherung geführt, die bei der Entwicklung komplexer Softwaresysteme hoher Qualität Hand in Hand gehen müssen:

*Anlehnung an
ISTQB®-Standard*

*Konstruktive und
analytische
Qualitätssicherung gehen
Hand in Hand.*

- Die konstruktive Qualitätssicherung versucht das Entstehen von Fehlern durch geeignete Maßnahmen während der Entwicklung zu verhindern. Sie basiert auf dem Einsatz von Methoden, Konstruktionsprinzipien, formalen Verfahren, Entwicklungswerkzeugen und Vorgehensmodellen.
- Die analytische Qualitätssicherung versucht, entstandene Fehler möglichst früh zu entdecken und die erreichte Qualität zu bewerten. In diese Kategorie fallen insbesondere die Testaktivitäten. Im Mittelpunkt steht die Frage, ob bestimmte vorgegebene Qualitätskriterien erfüllt bzw. Qualitätsziele erreicht werden.

Das Glossar des International Software Testing Qualifications Board (ISTQB[®], vgl. [ISTQB 2015]) gibt die folgende Definition des Begriffes »Testen«:

Definition

Testen

Der Prozess, der aus allen Aktivitäten des Lebenszyklus besteht (sowohl statisch als auch dynamisch), die sich mit der Planung, Vorbereitung und Bewertung eines Softwareprodukts und dazugehöriger Arbeitsergebnisse befassen. Ziel des Prozesses ist sicherzustellen, dass diese allen festgelegten Anforderungen genügen, dass sie ihren Zweck erfüllen, und etwaige Fehlerzustände zu finden. [URL: ISTQB]

Man unterscheidet demnach bei der analytischen Qualitätssicherung

- statische Tests (insb. Reviews, aber auch werkzeuggestützte Analysen), die den Prüfgegenstand (Teil-, Zwischen- oder Endprodukt, im Folgenden als Testobjekt bezeichnet) nicht ausführen und somit auf alle Entwicklungsprodukte, also auch z.B. Anforderungs- und Entwurfsspezifikationen, anwendbar sind, und
- dynamische Tests, die »ausführbare« Testobjekte, in der Regel also Programmcode, erfordern und diese mit dem Ziel ausführen, Fehler zu finden.

Dynamischer Test

Beim dynamischen Softwaretest – oft einfach als »Testen« bezeichnet – werden ausführbare Testobjekte, d.h. Programmteile, ganze Programme oder Systeme (system under test, SUT), auf systematische Weise mit dem Ziel ausgeführt, die korrekte Umsetzung der Anforderungen nachzuweisen, Fehlerwirkungen aufzudecken und das »Vertrauen« in das Testobjekt zu erhöhen. Vor der Testdurchführung werden für ausgewählte Eingaben anhand der Testbasis wie z.B. der Spezifikation des Prüflings die erwarteten Ausgaben ermittelt. Nach jeder Testdurchführung wird dann die tatsächliche Ausgabe mit der erwarteten verglichen.